

THE PROBABILISTIC ANALYSIS OF THE NETWORK GRAPH CREATED BY DYNAMIC BOUNDARY COVERAGE

GANESH P. KUMAR ^{*†} AND SPRING BERMAN ^{†‡}

Abstract.

Key words.

AMS subject classifications.

1. Introduction.

1.1. Background. We address the problem of achieving *boundary coverage* with a swarm of autonomous robots. In this task, a group of robots must allocate themselves around the boundary of a region or object according to a desired configuration or density. We specifically consider problems of *dynamic* boundary coverage, in which robots asynchronously join a boundary and later leave it to recharge or perform other tasks. Applications: mapping, exploration, environmental monitoring, surveillance, disaster response tasks such as cordoning off hazardous areas; collective payload transport, in which the group cooperatively transports a load to a destination, for automated manipulation, assembly, construction, and manufacturing.

We focus on *stochastic coverage schemes (SCS)*, in which robots probabilistically choose positions on the boundary. Our interest in SCS, as opposed to deterministic coverage schemes, is motivated by the following reasons. First, they enable a probabilistic analysis of the graph for different classes of inputs identified by the joint pdf of robot positions. Second, SCS allow us to model natural phenomena such as *Random Sequential Adsorption (RSA)* [30], the clustering of ants around a food item [20], and *Renyi Parking* [6], the process by which a fleet of cars parks without collisions on a parking lot. Lastly, results from SCS allow us to analyze the distributions of robots with noisy sensing and actuation, even though the underlying coverage scheme may be deterministic.

1.1.1. Assumptions about Robot Capabilities. We assume that each robot can locally sense its environment and communicate with other robots nearby. Disk model of sensing/communication. Robots can distinguish between other robots and a boundary of interest. The robots lack global localization: highly limited onboard power may preclude the use of GPS, or they may operate in GPS-denied environments. The robots also lack prior information about their environment. Each robot exhibits random motion that may be programmed, for instance to perform probabilistic search and tracking tasks, or that arises from inherent sensor and actuator noise. This random motion produces uncertainty in the locations of robot encounters with a boundary. For this reason, we refer to the task as *stochastic boundary*

[†]School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe AZ 85287.

[‡]School for Engineering of Matter, Transport and Energy, Arizona State University, Tempe AZ 85287.

[¶]This work was supported by DARPA Young Faculty Award no. D14AP00054.

coverage. In addition, we assume the robots have sufficient memory to store certain data structures.

1.2. Summary of Results. We devise a data structure to implement our coverage schemes, and we compute the probabilities of connectivity of various coverage schemes.

1.3. Models of Boundary Coverage. We consider a team of robots, $\{\mathcal{R}_i\} \in [1 \dots n]$ in a bounded environment \mathcal{E} . Robots are provided only with their (perfect) odometric readings and Wifi measurements, and a camera for detecting landmarks. Each robot is a disk of diameter R , and its Wi-fi has a coverage radius of d . They have no knowledge of their global positions or other means to localize. In the environment is placed a load in the form of a thin line, called the *Boundary* \mathcal{B} , which is colored black, distinctively from the rest of the environment. One endpoint of \mathcal{B} is painted white.

Since the main thrust of this paper lies in the randomized analysis of the network created by the robots, we will make the following simplifying assumptions. All robots are synchronized in time, with respect to a global clock. No robot fails in the course of its execution. Dealing with failures, and determining the success of boundary coverage in rugged environments where Wifi may fail are issues to be addressed in our future work.

To begin with, we consider point robots, for which $R = 0$ and thus the issue of inter-robot collisions does not arise. Let n of them attached to \mathcal{B} at a time instant $t \in \mathbb{N}$ [18]. Let the position of robot i be x_i . Define the vector of *unordered* positions to be $\mathbf{x}(t) := [x_{1 \dots n}]^T$. It will be convenient to make our computations if we sort this vector in nondecreasing order to get its permutation $\mathbf{x} = [x_{1 \dots n}]^T$, whose entry x_i is the i -th robot from the left, and not necessarily the position of R_i . Since x_i forms the i -th smallest of the n entries of \mathbf{x} , it is called the i -th *order statistic* of the positions [5]. We may think of \mathbf{x} (and \mathbf{x}) as the realization of a PP in \mathcal{B}^n , so that \mathbf{x} forms a point in \mathcal{B}^n . Define the random variable associated with x_i to be X_i , and place all these rv's in a vector \mathbf{X} that defines the PP.

For convenience, we introduce two artificial robots $x_0 = 0$ and $x_{n+1} = s$ stationed at the endpoints of \mathcal{B} . Since connectivity deals with inter-robot distances, it helps to think of them directly rather than in terms of \mathbf{x} . Define the i -th *slack* s_i to be the distance from x_i to x_{i+1} , and the *slack vector* $\mathbf{s}_{1:n+1} := x_{1:n+1} - x_{0:n}$ to be the vector of all slacks. Analogous to the rv's associated with positions, define the rv's S_i and the vector \mathbf{S} . We may think of \mathbf{s} as a point in \mathcal{B}^{n+1} .

Now we introduce the notion of connectivity by defining a *communication range* $d \in [0, s]$. Two robots x_i and x_j are connected iff $|x_i - x_j| \leq d$. We model connectivity by a graph $\mathcal{G}(\mathbf{x})$, whose nodes are x_i (or x_i), and edges are formed by pairs of connected robots. Since each node is a geometric position, $\mathcal{G}(\mathbf{x})$ forms a *Geometric Graph*. We define a position vector \mathbf{x} to be connected iff $\mathcal{G}(\mathbf{x})$ has a path from x_0 to x_{n+1} , and disconnected otherwise. When robot positions are chosen randomly, this graph becomes a Random Geometric Graph (RGG) [27].

We will now make the transition from point robots to R -sized ones on the boundary. Define the *position* of a robot as that of its left end, so that robot R_i located at x_i occupies the interval $\mathcal{S}_0 i = [x_i, x_i + R]$. The support of an attached robot position is $\mathcal{B}' = [0, s - R]$ (or a subset of it), so that the robot does not fall of the boundary endpoint $x = s$. We define the position vector \mathbf{x} of n robots to be *feasible* if there are no interrobot collisions, i.e. each slack s_i is at least R .

When a robot attaches to the boundary, it selects an *interval* of the boundary of length R lying completely within the boundary. We will generally be able to abstract intervals into points,

and consequently think of a SCS as the choice of multiple random *points* on the boundary. Formally, a SCS is a *one-dimensional Point Process (PP)* [12] realized on the boundary. A special case of a PP involves robots attaching to a boundary at predefined locations. We will be interested chiefly in the *Poisson Point Process (PPP)* in which robots attach independently to the boundary, and its generalizations such as the *Markov Process* [12]. The independent attachments in PPPs make them easy to analyze; on the other hand, interactions between robots are harder to handle and require generalizations of PPPs.

To simplify our analysis, we will first work with point robots in Sec.3 which have $\Delta = 0$, and consequently preclude inter-robot collisions. Point robots are an idealization of *finite* robots which have nonzero diameter; they also provide useful approximations to the behavior of finite robots when $\Delta \ll s$. We will compute the *connectivity properties* of each SCS that we address, which include the probability of saturation, the distribution of distances to the nearest neighbor, and the joint and marginal pdfs of robot positions and inter-robot distances.

1.4. Problem Statement. We require the robots to perform the following tasks:

PROBLEM 1.

1. *Form a connected network at the white endpoint of \mathcal{B} .*
2. *Attach to the boundary, forming a connected network, or cover as much of the boundary as possible.*
3. *Efficiently the list of positions taken up by the team on \mathcal{B}*
4. *Be able to update the map efficiently as robots join and leave the boundary*
5. *Determine at any point of time the network graph, including the following properties: coverage length, number of redundant robots (i.e. those that can be removed without loss of coverage).*

PROBLEM 2. *Compute the network properties for a random attachment scenario.*

1.5. Related Work.

1.5.1. Control of Multi-Robot Systems. Previous work on decentralized multi-robot boundary coverage has focused on controlling robots to converge to uniform or arbitrary formations on a circle [32]. In contrast to this work, we consider cases where there is inherent and/or programmed stochasticity in the robots' motion, and our objective is to achieve robot configurations with *target statistical properties*. We assume that every robot has minimal capabilities: no global position information, and sensing or communication only within a small radius. Task allocation strategies that are suitable for such scenarios often derive robot control policies from a continuum model of the swarm population dynamics, or *macroscopic model*, in order to enable the control policies to scale with the swarm size. Various stochastic approaches to robot task allocation have focused on optimizing the task-switching rates of such macroscopic models [2, 4, 21, 22, 26]. Macroscopic models have also been applied to problems of robotic assembly of products, as well as robotic self-assembly [9, 17, 23, 25].

1.5.2. Wireless Networks. Since our interest lies in getting a robot team to form a connected Wi-fi network around a boundary, we will dwell on data structures for routing in Mobile Ad Hoc Networks (*MANETs*) [15]. However, our main focus lies in analyzing the properties of the Geometric Graph (GG) formed by the multi-robot network. Our probabilistic analysis borrows heavily from the formalism of *Poisson Point Processes (PPP)* [28], a class

of spatial stochastic processes in which each robot takes positions independently of others. The network induced by a PPP is a *Random Geometric Graph (RGG)* [28]. When attachments are required to be *collision-free*, i.e. have no colliding pairs of robots, they are characterized by a Matern hard core process (*HCP*). These spatial processes and their resulting RGGs have been extensively used in the wireless communication literature [12].

1.5.3. Computational Geometry. All our results that compute $p_{\text{con}}(G)$, the connectivity probability of \mathcal{G} , involve the order statistical properties of the pdf governing the attachment of individual robots, called the *parent pdfs*, or parents for short [5]. The order statistics of a collection of random variables $\mathbf{X}_{1:n}$ are generated by sorting them in nondecreasing order to get their permutation \mathbf{X} . The order statistics of uniform iid parents are the easiest to analyze; moreover, nonuniform iid parents of other forms may be readily converted to their uniform counterparts using the *probability integral transform* [5]. We use the computations in [8] to derive a pseudo-P lower bound for computing $p_{\text{con}}(\mathcal{G})$ for uniform parents in ???. Further, we demonstrate that determining $p_{\text{con}}(\mathcal{G})$ for arbitrary iid pdfs is #PH, by reduction from a result in [7, 8].

The analogous computation of $p_{\text{con}}(\mathcal{G})$ for independent, non-identical (inid) pdfs is considerably more complicated, and uses the *Bapat-Beg theorem* [10]. This computation is governed by a [?]. We show in Sec. 5.3 by a reduction from the boolean permanent problem [?] that computing p_{con} is #PH.

Our results for the Stochastic Coverage Scheme (SCS) involving Renyi parking stems from the work of Renyi [29] and Dvoretzky et al. [6]. The Renyi Parking Problem (RPP) defined in Sec. 4 has been extensively studied in the physics literature under the name *Random Sequential Adsorption (RSA)*, the process by which molecules get adsorbed onto a substrate surface [30, 31]. The delay differential equation that governs the mean number of parked cars is extensively analyzed in [6, 30]; moreover, [30] computes the asymptotic properties of an interval tree that stores the occupied subintervals of the parking lot. To our knowledge, however, there has been no analysis of the spatial probability density functions (pdfs) generated by the RPP. We derive an algorithm for computing this pdf using results from order statistics [5].

2. Deterministic Coverage Strategy (DCS) for \mathcal{B} . We will first provide a DCS for a group of finite-sized robots to form a connected network with uniform inter-robot spacing along a boundary \mathcal{B} . This algorithm starts with a simple procedure, detailed in Algorithm 1 below, that is guaranteed to make all robots join the same network. Assuming there are no faults, Algorithm 1 will terminate with all robots joining the network created by robot \mathcal{R}_1 .

In this MANET, *every* node acts as a router. Once the robot team forms a connected network after the execution of Algorithm 1, the ID of every robot in the network is determined by flooding. This set of IDs is stored in the routing table of every robot. Subsequently, one robot (say, \mathcal{R}_1) leaves the network to determine the length s of the boundary using its odometry, and then rejoins the network by following the boundary back to its white endpoint. The maximum number of robots that can possibly attach to \mathcal{B} is

$$(2.1) \quad n_{\text{max}} = \lfloor \frac{s-R}{R} \rfloor.$$

The minimum number of robots required to ensure connectivity is

$$(2.2) \quad n_{\text{min}} = \lfloor \frac{s}{d} \rfloor.$$

Algorithm 1

```
1: procedure FORM A CONNECTED NETWORK( $i, n$ )
2:   state  $\leftarrow$  EXPLORE
3:   while Black Boundary not seen do
4:     Execute Lawnmower walk
5:   while White endpoint not seen do
6:     Traverse  $\mathcal{B}$ 
7:   if  $id = 1$  then
8:     Create Wifi network
9:   else
10:    while network not created do
11:      Wait
12:    Join Wifi network
13:   state  $\leftarrow$  CONNECTED
```

Based on these limits, the connectivity of the robot network falls into three categories:

Case 1: If $n < n_{\min}$, then at most nd of the boundary can be covered.

Case 2: If $n \in [n_{\min}, n_{\max}]$, then all robots can be accommodated, and can cover the boundary entirely.

Case 3: If $n > n_{\max}$, then $n - n_{\max}$ robots have to be dropped from coverage. In this case, the first n_{\max} robots attach to the boundary, and the remaining are dropped.

Robots subsequently take up positions that are spaced d apart, so that $x_i = (i - 1)d$, using their odometry, with the white endpoint being considered $x_0 = 0$. Afterwards, the robots can coordinate to attach to, or detach from, \mathcal{B} . This DCS can be easily adapted to any SCS as follows. Instead of taking up equidistant positions, the robot team collectively samples from a joint pdf of their positions on \mathcal{B} , and attaches to these positions in order. The initial step of forming a connected network makes it easy to execute either coverage scheme.

2.1. Determining properties of \mathcal{G} . The robots can use the *Optimized Link State Routing (OLSR)* protocol [13, 14] to determine the connectivity, coverage length, and number of edges of \mathcal{G} at any instant. OLSR is a proactive, table-driven routing protocol, each of whose nodes maintains a table of 1-hop neighbors, which are found by flooding HELLO messages through the network. When a new node joins or an existing one leaves the network, a set of TC (Topology Control) messages are initiated by the neighbors of this node, flooding the network with updated routing tables. Robots can determine the properties of the network as follows:

1. **Decide network connectivity:** Every robot floods the network with a message consisting of its id and its position. The flooding of the network is deemed to stop after a timeout τ , known to all robots, at which time every robot compiles a table of robot positions and id's. From this table, the leftmost and rightmost robot IDs, x_1 and x_n , are identified. If $x_i \leq d$ and $x_n \geq s - d$, then the entire network is connected. Otherwise, each robot deems the network to be disconnected as a whole.
2. **Number of Connected Components:** Each robot can determine its own connected component from the routing table. If the connected component of any robot covers both end-points of \mathcal{B} , then the network as a whole is connected. Otherwise, after

a timeout period τ that is known to all robots, one robot (say \mathcal{R}_1) detaches and traverses the boundary, querying the nearest robots about their connected components. After one full traversal of \mathcal{B} , \mathcal{R}_1 computes the total number of connected components and updates. It subsequently updates other robots of this number in a second traversal of \mathcal{B} .

3. **Number of edges:** This is a variant of the approach for computing the number of connected components. Each connected component may determine the number of edges in it independently of others. As before, \mathcal{R}_1 detaches and traverses \mathcal{B} to query the number of edges in each connected component, which is then broadcast to each robot.

2.2. Creating and updating a list of robot positions. Each robot \mathcal{R}_i in the network maintains data about robot positions along the boundary in the form of an *interval tree* [1]. If this position data is too large to fit into the memory of a robot, it will keep track only of its m nearest neighbors, where the size m is the maximum allowable size of the tree. The interval tree handles insertions, deletions and search queries in $O(\log m)$ time.

An incoming robot that wishes to attach to \mathcal{B} , say \mathcal{R}_{n+1} , will approach \mathcal{B} and send a broadcast query to the network to determine the locations of slacks that are large enough for it to attach. A subset of the attached robots will then respond to \mathcal{R}_{n+1} with a list of slacks where it may attach. Subsequently, \mathcal{R}_{n+1} attaches and broadcasts its position to its neighbors, who in turn update their position data. Likewise, an outgoing robot \mathcal{R}_1 notifies its neighbors of its impending detachment. The neighbors recompute the resulting slacks, making note of any disconnected slacks introduced by the detachment of \mathcal{R}_1 . They subsequently clear \mathcal{R}_1 for detachment, following which \mathcal{R}_1 detaches.

2.3. Discussion. This section has presented only a high level view of the BC protocol. We have deliberately processor failures, asynchrony, and anonymity for the sake of simplicity. A detailed discussion of these issues would distract from our objective of analyzing \mathcal{G} .

3. IID Coverage by Homogeneous Point Robots. In this section, we consider an SCS driven by a Poisson Point Processes (PPP), in which every robot attaches independently to \mathcal{B} , following the same spatial parent pdf. In other words, \mathbf{X} consists of iid random variables, and defines a PPP on \mathcal{B} . Specifically, suppose that the parent pdf and cdf are $p_f(x)$ and $p_F(x)$ respectively, both supported on \mathcal{B} . Then the number of points N falling on a subinterval $[a, b]$ of \mathcal{B} is a Poisson random variable with underlying pdf $p_f(x)$:

$$(3.1) \quad N(a, b) \sim \text{Poi}(\lambda) \text{ where } \lambda = p_F(b) - p_F(a).$$

We derive connectivity results for this SCS for a fixed team of n robots and then generalize these results to a case of dynamic attachment and detachment. Our primary parameters of interest are the connectivity properties of $\mathcal{G}(\mathbf{X})$, namely the probability p_{con} of connectivity, the expected degree of a vertex, and the number of clusters, all of which. Subsequently, we determine the *spatial pdfs* of \mathbf{X} and \mathbf{S} , both for connected and unconnected components of \mathcal{G} . We then apply these results to analyze the *temporal properties*, such as recurrence times, of dynamic scenarios in which robots attach and detach probabilistically. [18, 19].

3.1. Geometric interpretation of connectivity. We interpret \mathbf{x} and \mathbf{s} as points in \mathbb{R}^n and \mathbb{R}^{n+1} , respectively. The entries of \mathbf{x} are nondecreasing, and thereby define the *position*

simplex [19]

$$(3.2) \quad \mathcal{P} = \{\mathbf{x}_{1:n} : \text{for all } i : 1 \leq i \leq n, \text{ we have that } 0 \leq x_i \leq x_{i+1} \leq s\}.$$

Likewise, all valid slack vectors, i.e. those that arise from a robot configuration on \mathcal{B} , have entries whose sum is the boundary length s . Geometrically, \mathbf{s} defines a point on a simplex \mathcal{S} that we call the *slack simplex* [18], given by

$$(3.3) \quad \mathcal{S} := \{\mathbf{s} : \mathbf{1}^T \mathbf{s} = s, \text{ and } \mathbf{0} \leq \mathbf{s} \leq s\mathbf{1}\} = s \cdot \Delta_n,$$

where

$$(3.4) \quad \Delta_n := \{\mathbf{s} : \mathbf{1}^T \mathbf{s} = 1, \text{ and } \mathbf{0} \leq \mathbf{s} \leq \mathbf{1}\}$$

is the canonical simplex in \mathbb{R}^n . The vertices of \mathcal{S} are

$$(3.5) \quad \mathbb{V}(\mathcal{S}) = s \cdot \mathbb{I}_{n+1} = s \cdot [\hat{e}_1 \dots \hat{e}_{n+1}],$$

where \hat{e}_i is the unit vector along the i -th axis. Eq.(3.3) expresses \mathcal{S} as a degenerate simplex, with Lebesgue measure zero in \mathbb{R}^{n+1} . For our computations, we will need to express \mathcal{S} in full-dimensional form as

$$(3.6) \quad \mathcal{S} = \{\mathbf{s}_{1:n} \in \mathbb{R}^n : \mathbf{0}^T \leq \mathbf{s}, \text{ and } \mathbf{1}^T \mathbf{s} \leq s\},$$

by dropping the last slack s_{n+1} , which is determined by its predecessors. Observe that all connected configurations, regardless of whether they are valid, fall within the hypercube

$$(3.7) \quad \mathcal{H} := \{0 \leq \mathbf{s} \leq d\mathbf{1}^T\} = d \cdot \mathcal{C}_n,$$

where \mathcal{C}_n is the unit hypercube $[0, 1]^n$.

A valid slack vector has to lie in the intersection $\mathcal{S} \cap \mathcal{H}$ to represent a connected configuration. Define the *connected region* as $\mathcal{F} = \mathcal{S} \cap \mathcal{H}$ and the *disconnected region* as $\mathcal{U} := \mathcal{S} \setminus \mathcal{F}$. We show in [18] that d falls into three ranges,

$$(3.8) \quad d \in \begin{cases} [0, \frac{s}{n+1}], \text{ for which } \mathcal{F} = \emptyset \text{ and } p_{\text{con}} = 0; \\ (\frac{s}{n+1}, 1), \text{ for which } \mathcal{F} \subsetneq \mathcal{S} \text{ and } p_{\text{con}} \in (0, 1); \\ [1, \infty), \text{ for which } \mathcal{F} = \mathcal{S} \text{ and } p_{\text{con}} = 1. \end{cases}$$

We may express \mathcal{F} as

$$(3.9) \quad \mathcal{F} = \{\mathbf{s}_{1:n} \in \mathcal{S} : s - d \leq \mathbf{1}^T \mathbf{s}_{1:n} \text{ and } \mathbf{s} \leq d\mathbf{1}^T\}.$$

The parent p_f generates the joint pdf [5]

$$(3.10) \quad f_{\mathbf{x}}(\mathbf{x}) = n! \prod_{1 \leq i \leq n} f(x_i) \mathbf{I}_{\mathcal{P}}.$$

over the the position simplex, where \mathbf{I} denotes the *indicator function* over the region in its subscript. This pdf is called the Janossy pdf [12] of the PPP. Changing the argument from \mathbf{x} to \mathbf{s} gives us

$$(3.11) \quad f_{\mathbf{s}}(\mathbf{s}_{1:n}) = n! \prod_{1 \leq i \leq n} f(\sum_{j=1}^i s_j) \mathbf{1}_{\mathcal{S}}.$$

We will compute the properties of \mathcal{G} in the coming subsections, starting with p_{con} . The formula for finding p_{con} provides us with a template for partitioning \mathcal{S} into regions that are amenable to computing the following properties of \mathcal{G} : the number of connected components of \mathcal{G} , the coverage induced by \mathcal{G} , and the edge count of \mathcal{G} . While these quantities are nontrivial to compute for RGGs of arbitrary dimension [27], there exist straightforward, if tedious, algorithms to compute them for a single dimension. All these algorithms essentially involve computing the ratio of integrals of the joint pdf $\tilde{\mathbf{s}}$ over a subset of \mathcal{S} .

3.2. Probability of connectivity. The probability of connectivity p_{con} is the ratio of the volume of the joint pdf lying over \mathcal{F} to that over \mathcal{S} :

$$(3.12) \quad p_{\text{con}} := \frac{\text{Leb}(\mathbf{S}, \mathcal{F})}{\text{Leb}(\mathbf{S}, \mathcal{S})} = \frac{\int_{\mathcal{F}} f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}}{\int_{\mathcal{S}} f_{\mathbf{S}}(\mathbf{s}) d\mathbf{s}}.$$

where $\text{Leb}(\mathbf{S}, \mathcal{S})$ computes the Lebesgue measure of the joint pdf of \mathbf{S} over \mathcal{S} . The denominator is relatively easy to evaluate analytically using barycentric coordinates [11], while the integral over \mathcal{F} is harder to compute, since there is no obvious way to decompose it into simplices. A naive algorithm that triangulates \mathcal{F} into simplices will take a long time in practice when n is large. Instead, we may write $\text{Leb}(\mathbf{S}, \mathcal{F}) = \text{Leb}(\mathbf{S}, \mathcal{S}) - \text{Leb}(\mathbf{S}, \mathcal{U})$, decompose \mathcal{U} into simplices rather than \mathcal{F} [19], and finally compute $p_{\text{con}} = 1 - \frac{\text{Leb}(\mathbf{S}, \mathcal{U})}{\text{Leb}(\mathbf{S}, \mathcal{S})}$.

This decomposition of \mathcal{U} will result in *overlapping* simplices, whose measures we can combine using the combinatorial approach described in [19].

$$(3.13) \quad \mathcal{U} = \bigcup_{\mathbf{v} \in \{0,1\}^n \setminus \{\mathbf{0}\}} \mathcal{U}(\mathbf{v}),$$

where $\mathcal{U}(\mathbf{v})$ forms a simplex of side $(s - d\mathbf{1}^T \mathbf{v})$, with the vertices

$$(3.14) \quad \mathbb{V}(\mathcal{U}(\mathbf{v})) = (s - d\mathbf{1}^T \mathbf{v}) \mathbb{I}_{n+1} + \mathbf{v}.$$

This expression is nonnegative when $s \geq d\mathbf{1}^T \mathbf{v}$, so that only those vertices with at most $n_{\min} = \lfloor \frac{s}{d} \rfloor$ d 's in them need be considered. The value n_{\min} is the minimum number of robots required for connectivity, as well as the maximum possible number of disconnected slacks. We call the simplex $\mathcal{U}(\mathbf{v})$ the *compatible simplex* of \mathbf{v} . Compatible simplices overlap, so the sum of their measures exceeds that of \mathcal{U} . We first decompose \mathcal{U} using the inclusion-exclusion principle (IEP) as:

$$(3.15) \quad \mathcal{U} = \bigcup_{\text{odd } \mathbf{v}} \mathcal{U}(\mathbf{v}) \setminus \bigcup_{\text{even } \mathbf{v}} \mathcal{U}(\mathbf{v})$$

where the (even or odd) parity of \mathbf{v} is that of its number of 1-bits. We immediately have

$$(3.16) \quad \text{Leb}(\mathbf{S}, \mathcal{U}) = \sum_{\mathbf{v} \in \{0,1\}^{n+1}} (-1)^{\mathbf{1}^T \mathbf{v}} \text{Leb}(\mathbf{S}, \mathcal{U}(\mathbf{v})).$$

Our remaining computations will rely heavily on the decomposition of \mathcal{S} into $\mathcal{U}(\mathbf{v})$.

3.3. Number of Components. A slack vector \mathbf{s} has a single connected component iff it is connected, i.e if $\mathbf{s} \in \mathcal{F}$. Each unsaturated slack in \mathbf{s} inserts a new connected component into \mathcal{G} . Define the component counting function

$$(3.17) \quad \text{cmp} : \mathcal{S} \mapsto \mathbb{N} \text{ with } \text{cmp}(\mathbf{s}) = \sum_i^{n+1} \begin{cases} 1 & \text{if } s_i \leq d \\ 0 & \text{otherwise.} \end{cases}$$

where $\mathbb{N} = \{0, 1, 2, \dots\}$. By definition, we have $\text{cmp} = n + 1 - \mathbf{1}^T \mathbf{v}$ identically over $\mathcal{U}(\mathbf{v})$. We may then compute the expectation of cmp over \mathcal{S} by writing $\mathcal{S} = \mathcal{U} \cup \mathcal{F}$, and consequently get

$$(3.18) \quad \mathbb{E}(\text{cmp}) = \text{Leb}(\text{cmp}, \mathcal{S}) = \sum_{\mathbf{v} \in \{0,1\}^{n+1} : \mathbf{1}^T \mathbf{v} \leq n_{\min}} (-1)^{\mathbf{1}^T \mathbf{v}} \int_{\mathcal{U}(\mathbf{v})} (n + 1 - \mathbf{1}^T \mathbf{v}) f_{\mathbf{s}}(\mathbf{s}) d\mathbf{s}.$$

3.4. Coverage Length. To determine the length of the \mathcal{B} covered by \mathbf{s} , we will introduce the *coverage function* $\text{cov}(\mathbf{s})$. If \mathbf{s} is connected, then its coverage length $\text{cov}(\mathbf{s})$ is the boundary length s . If \mathbf{s} has a disconnected slack s_i , a length of $s_i - d$ is left without coverage. This motivates us to define cov by

$$(3.19) \quad \text{cov} : \mathcal{S} \mapsto \mathbb{R} \text{ with } \text{cov}(\mathbf{s}) = s - \sum_i^{n+1} \max(s_i - d, 0).$$

Computing $\mathbb{E}(\text{cov})$ over \mathcal{S} does not get simplified by the decomposition $\mathcal{S} := \mathcal{U} \cup \mathcal{F}$, for cov is non-constant over \mathcal{U} . A straightforward integration gives us

$$(3.20) \quad \mathbb{E}(\text{cov}) = \text{Leb}(\text{cov} \cdot \mathbf{S}, \mathcal{S}) = s \cdot \text{Leb}(\mathbf{S}, \mathcal{S}) - \sum_{i=1}^{n+1} \max(0, s_i - d) f_{\mathbf{s}}(\mathbf{s}) d\mathbf{s}.$$

3.5. Number of edges of \mathcal{G} . We will define the edge counting function $\text{edg}(\cdot)$ over positions rather than slacks. Given the position vector \mathbf{x} , there exists an edge between \mathbf{x}_i and \mathbf{x}_j iff $\mathbf{x}_j - \mathbf{x}_i \leq d$. Accordingly, we have

$$(3.21) \quad \text{edg} : \mathcal{P} \mapsto \mathbb{N}, \text{ with } \text{edg}(\mathbf{x}) := \sum_{i=1}^{n-1} \sum_{j=i+1}^n 1 - \max(\mathbf{x}_j - \mathbf{x}_i - d, 0)$$

with $\mathbb{E}(\text{edg})$ being the integral of Eq.(3.21) over \mathcal{P} :

$$(3.22) \quad \mathbb{E}(\text{edg}) = \text{Leb}(\mathbf{X}, \mathcal{P}) - \int_{\mathcal{P}} \sum_{i,j:1 \leq i < j \leq n} \max(\mathbf{x}_j - \mathbf{x}_i) f_{\mathbf{x}}(\mathbf{x}) d\mathbf{x}.$$

3.6. Dynamic coverage with iid attachment. Now we will examine a strategy in which the robot team dynamically attaches and detaches from the boundary, with their spatial attachment pdfs being iid on \mathcal{B} .

3.6.1. Dynamic attachment. We first consider the case in which robots attach to the boundary without detaching. One robot position is chosen at every time step using the parent pdf p_f until connectivity is achieved. We compute the expected time until connectivity, or the expected *stopping time* of the SCS. To determine the stopping time, we consider the sequence $(p_i)_{i \in \mathbb{N}}$, where p_i is the probability of connectivity with i robots. Irrespective of the parent pdf p_f , having more robots on \mathcal{B} leads to a greater probability of connectivity. Consequently, the sequence (p_i) is monotonically increasing on the support $[n_{\min}, \infty)$ and tends to unity as i grows without bound. We also know that $p_{i:i \leq n_{\min}} = 0$. Consequently, the attachment process will terminate (resp. fail to terminate) at $i \geq n_{\min}$ robots with probability p_i (resp. $1 - p_i$). The probability of connectivity being attained at i robots is:

$$(3.23) \quad \tau_i = \begin{cases} 0 & \text{for } 1 \leq i < n_{\min} \\ (1 - p_{i-1})p_i & \text{for } i \geq n_{\min} \end{cases}$$

Thus, τ_i is a generalized geometric random variable whose probability of success in a trial is distinct from that in its previous one. The expected stopping time is

$$(3.24) \quad \mathbb{E}(\tau) = \sum_{i=n_{\min}}^{\infty} i \tau_i.$$

Since $p_i > p_{n_{\min}}$ for $i > n_{\min}$, we expect quicker connectivity than that of a geometric random variable whose parameter is $p_{n_{\min}}$ $\mathbb{E}\tau \leq \frac{1}{p_{n_{\min}}}$.

3.6.2. Stopping time of connectivity for Uniform parent. The uniform parent has the special property that \mathbf{S} is jointly uniform over \mathcal{S} , with each slack being identically distributed (though not iid) as scaled exponentials of the form $s \cdot \text{Exp}(1)$. Further, the *order statistics* of the slacks, represented by the vector \mathbf{s} , formed by sorting \mathbf{s} in increasing order, obey the relations [5]:

$$(3.25) \quad \mathbb{E}(S_i) = \frac{s}{n+1} \sum_{j=1}^{n+1} \frac{1}{j} = \frac{s}{n+1} (H_{n+1} - H_i)$$

$$(3.26) \quad \mathcal{V}(S_i) = \sum_{j=i}^{n+1} \frac{1}{j^2}$$

where H_n denotes the harmonic numbers. The longest slack S_{n+1} has the expected value $\frac{sH_{n+1}}{n+1}$. To have $s_{n+1} \leq d$, we need $\frac{H_{n+1}}{n+1} \leq \frac{d}{s}$, which may be solved numerically to get the expected hitting time of \mathcal{F} . We may also estimate n if n is large by approximating H_n with $\log n$, providing

$$(3.27) \quad \frac{\log(n+1)}{n+1} \leq \frac{d}{s} \implies n = \exp(-W(\frac{d}{s})) - 1,$$

where W is the *Lambert W function*.

3.6.3. Dynamic attachment and detachment. We now extend the results in Sec.3.6 to a scenario in which we require robots to strike a balance between forming a connected network on the boundary and exploring the surrounding environment of the boundary. Formally, we are given that at every time instant $t \in \mathbb{R}_+$, a robot may be either attached to

the load or detached from it; in other words, the robot has a temporal state alphabet $\Sigma := \{A \text{ (attached)}, D \text{ (detached)}\}$.

PROBLEM 3. *Design the rates of switching between states, with a guarantee on the expected amount of time that the boundary will have a connected network.*

To analyze the behavior of the robots, we introduce the temporal state $\mathbf{N}(t) = [N_A(t) \ N_D(t)]^T$, whose entries denote the number of robots in states A and D , respectively. We assume that the total number of robots is conserved, which implies that

$$(3.28) \quad N_A(t) + N_D(t) = N_A(0) + N_D(0).$$

Now we suppose that robots change state per the chemical reactions



where r_{ij} , the reaction rate constant, is the probability per unit time of a robot in state i to switch to state j . The populations of robots in both states evolve over time as

$$(3.30) \quad \frac{d}{dt}\mathbf{N}(t) = \begin{bmatrix} -r_{AD} & r_{DA} \\ r_{AD} & -r_{DA} \end{bmatrix} \mathbf{N}(t).$$

At equilibrium, $\frac{d}{dt}\mathbf{N}(t) = \mathbf{0}$, and so Eq.(3.30) yields

$$(3.31) \quad \frac{N_A^*}{N_D^*} = \frac{r_{DA}}{r_{AD}}.$$

We solve for N_A^* and N_D^* using Eq.(3.28) and Eq.(3.31).

4. Uniform Coverage by Homogeneous Finite Robots. We now consider SCS with *finite robots*, each of which has a nonzero diameter R . Unlike the case of point robots, the *maximum number* of attached robots is finite and given by

$$(4.1) \quad n_{\max} = \lfloor \frac{s}{R} \rfloor.$$

Collision-free positions are a realization of a *Matern hard-core PP* [12], which prohibits its points from lying within a threshold distance of each other. The valid range for n is $[n_{\min}, n_{\max}]$. When $d \leq R$, every feasible configuration becomes a saturated one, causing n_{\min} to coincide with n_{\max} . The case $d = R$ is of special interest to us, since it is an instance of Renyi's Parking Problem.

PROBLEM 4. *Renyi's Parking Problem [6, 30] Cars of unit length park uniformly randomly on a segment of length s , avoiding collisions, until no parking space is available for the next car. Analyze the pmf of the final number of parked cars, \mathbf{N} .*

The mean number of parked cars, $\mathbb{E}\mathbf{N}$, obeys a delay integral equation with the asymptotic solution

$$(4.2) \quad \lim_{s \rightarrow \infty} \mathbb{E}\mathbf{N} = n_{pc} \cdot s \approx 0.748s.$$

where n_{pc} is Renyi's parking constant [29]. This result implies that we expect 75% of the segment to be occupied by cars at the point where there is no more room to accommodate another car. An exact solution for $\mathbb{E}\mathbf{N}$ leads to an intractable $\lfloor s \rfloor$ -dimensional integral [24]. Our SCS with fixed n and uniformly random attachments is a special case of Renyi's Parking Problem in which \mathbf{N} is trivial to compute. However, to our knowledge, there has been no analysis of the spatial pdfs that are generated by the parked cars in this problem, which we provide in Sec.4.1.

4.1. Connectivity of Collision-Free Parking. We now formulate the CF equivalent of the point-robot attachment in Sec.3. Define the *position* of a robot as that of its left end, so that robot R_i located at x_i occupies the interval $[x_i, x_i + R]$. The support of all attached robot positions is $\mathcal{B}' = [0, s - R]$, which ensures that no robot extends beyond the boundary endpoint $x = s$. We introduce two artificial robots at $x_0 = -R$ and $x_{n+1} = s$. Define \mathbf{x} to be *collision-free (CF)* iff

$$(4.3) \quad 0 \leq x_0, \quad x_n \leq s - R, \quad \text{and } x_i - x_{i-1} \geq R, \text{ for } i = 1 \dots n,$$

and define \mathcal{P}_{CF} to be the set of CF position vectors. Likewise define the CF subset of \mathcal{S} and the resulting favorable region by

$$(4.4) \quad \mathcal{S}_{\text{CF}} := \{\mathbf{s} \in \mathcal{S} : \mathbf{R} \cdot \mathbf{1}^T \leq \mathbf{s}\}$$

$$(4.5) \quad \mathcal{F}_{\text{CF}} := \mathcal{S}_{\text{CF}} \cap \mathcal{H} = \{\mathbf{s} \in \mathcal{S} : \mathbf{R} \cdot \mathbf{1}^T \leq \mathbf{s} \leq d \cdot \mathbf{1}^T\}.$$

Geometrically, \mathcal{S}_{CF} is a simplex with the hypercuboids $\mathbf{0} \leq s_i \leq R$ removed. Reasoning as in Sec.3, we have $p_{\text{con}}(\mathcal{G}) = \text{Vol}(\mathcal{F}_{\text{CF}})/\text{Vol}(\mathcal{S}_{\text{CF}})$; however, we are unable to simplify this formula further as we did there. The lack of a simplifying expression for p_{con} means that the computation of $p_{\text{con}}(\mathcal{G})$ has to involve the triangulation of \mathcal{F}_{CF} into simplices, a time-consuming operation that we explicitly avoided in Eq.(3.13). Likewise, expressions for the order statistics and slacks of CF positions are obtained by integrating the uniform joint pdf over \mathcal{S}_{CF} instead of \mathcal{S} , as do the formulae for the properties of \mathcal{G} .

5. Complexity Results of SCS.

5.1. Computing p_{con} for uniform iid parents. We now investigate the complexity of exactly computing the integrals in Eq.(3.16). We begin with a pseudo-P lower bound for computing $\text{Vol}(\mathcal{F})$, and consequently PCON for the uniform parent. We will then discuss lower bounds for non-uniform parents. Define the complexity theoretic problem $\text{PCON}(f, s, d, n) \mapsto p_{\text{con}}$, with

Input: Parameters of SCS : encoding of $pf, ; s, d \in \mathbb{Q}_+ ; n \in \mathbb{N}$

Output: Probability of connectivity $p_{\text{con}} \in \mathbb{Q}_+$

Rational inputs and outputs are specified as exact reduced fractions; for example s is input as the pair $(\text{num}(s), \text{den}(s))$. Let $\text{PCON}(\text{U})$ denote that subproblem of PCON over a uniform SCS.

THEOREM 5.1. $\text{PCON}(\text{U})$ can be solved in $\Omega(n)$ and $O(n \log n)$ time.

Applying Eq.(3.16) gives us [18]

$$(5.1) \quad \text{Vol}(\mathcal{S}) = \frac{s^n \cdot \sqrt{n+1}}{n!},$$

$$(5.2) \quad \text{Vol}(\mathcal{U}) = \sum_{k=1}^{n_{\min}} (-1)^{k-1} \binom{n+1}{k} \frac{(s - kd)^n \sqrt{n+1}}{n!},$$

$$\text{PCON} = \frac{\text{Vol}(\mathcal{F})}{\text{Vol}(\mathcal{S})} = 1 - \sum_{k=1}^{n_{\min}} (-1)^{k-1} \binom{n+1}{k} \left(1 - \frac{kd}{s}\right)^n.$$

Proof. Upper bound: Eq.(5.1) is a possible solution for $\text{PCON}(\text{U})$; therefore, its worst-case running time forms an upper bound for $\text{PCON}(\text{U})$. Eq.(5.1) runs in $O(n_{\min} \log n)$ time; its

worst-case instances, which have $n > n_{\min}$ take $O(n \log n)$ time, which forms an upper bound for any solution to PCON. Note that Eq. (5.1) forms a pseudo-P algorithm for PCON.

Lower bound: Consider PCON(U) instances with $s = \frac{4}{3}d$, for which

$$(5.3) \quad \text{PCON} = 1 - (n+1) \cdot \frac{1}{4^n} \approx 1 - \frac{n}{4^n}$$

The input size for these instances is $O(\log n)$, but the output size is exponential in the input size, implying an $\Omega(n)$ lower bound for any algorithm for PCON(U). The upper bound of $O(n \log n)$ is off from the lower bound $\Omega(n)$ only by a polynomial in the input size of PCON(U), implying that algorithms for PCON consume more time in outputting the solution rather than computing it. The lower bound is an exponential function of the input size, hence $\text{PCON(U)} \in \#\text{EXP}$. \square

Time complexity of computing $\text{Vol}(\mathcal{F})$: The \sqrt{n} in the formula for $\text{Vol}(\mathcal{F})$ makes it impossible to provide a bounded decimal expansion to $\text{Vol}(\mathcal{F})$, hence the complexity of writing down $\text{Vol}(\mathcal{F})$ is *infinite*, except when n is a square. Remedying this unbounded expansion requires us to compute $\text{Vol}(\mathcal{F}) \cdot \frac{n!}{\sqrt{n+1}}$, for which the same bounds as PCON(U) apply. The same bounds apply to $\text{Vol}(\mathcal{U}) \cdot \frac{n!}{\sqrt{n+1}}$.

5.2. Generalized Simplex Hypercube intersection. In the coming sections, we will demonstrate that our problems are #PH by reduction from the VHSP problem.

LEMMA 5.2. *Define the problem VHSP:*

Input: Parameters $\mathbf{a}_{1:n}, b$ of the halfspace $\mathcal{T} := \{\mathbf{s} \in \mathbb{R}^n : \mathbf{a}^T \mathbf{s}_{1:n} \leq b\}$, with \mathbf{a} and b are positive rationals

Output: Volume of intersection of \mathcal{T} with the unit hypercube $\mathcal{C} := [0, 1]^n$

The solution to VHSP given by

$$(5.4) \quad \text{Vol}(\mathcal{T} \cap \mathcal{C}) = n! \prod_{i=1}^n \sum_{\mathbf{v} \in [0, 1]^n} (-1)^{\mathbf{1}^T \mathbf{v}} \max((b - \mathbf{a}^T \mathbf{s})^n, 0).$$

is #PH. Equivalently, it is #PH to find the probability that a random point in \mathcal{C} satisfies a single linear inequality.

It will be useful to redefine VHSP as an intersection between a half-space with unit coefficients and a generic hypercuboid. Introduce the primed variables $s'_i := a_i s_i$, and note that VHSP asks for $\text{Vol}(\mathcal{T}' \cap \mathcal{C}')$, where

$$(5.5) \quad \mathcal{T}' := \{\mathbf{s}' \in \mathbb{R}^n : \mathbf{1}^T \mathbf{s}' \leq b\} \text{ and}$$

$$(5.6) \quad \mathcal{C}' := \prod [0, a_i].$$

5.2.1. Nonuniform iid parents. We will now give an example of a nonuniform parent whose p_{con} is #PH to compute. For this purpose, define a k -piecewise uniform (k -PWU) over a finite support $[0, L]$ as follows. Partition the support into k nonempty subintervals as

$$(5.7) \quad [0, L] := [0, L_1] \cup [L_1, L_2] \dots \cup [L_k, s].$$

On the i -th subinterval, $f(x)$ is defined to be the constant $p_i \in [0, 1]$, which are chosen to satisfy $\sum p_i(L_i - L_{i-1}) = 1$, and consequently f is a pdf on \mathcal{B} .

THEOREM 5.3. $\text{PCON}(n\text{U})$ is #PH.

Proof. Given the VHSP instance with the dimension $n + 1$, having the hypercuboid $\mathcal{C}' := \prod_{1 \leq i \leq n+1} [0, l_i]$ and the half-space sum b . Let $L := \sum l_i$. Define the equivalent instance of PCON to have the parameters

$$(5.8) \quad s := \frac{b}{L}, \quad d := 1, \text{ and } {}_p f := \frac{l_i}{L} \text{ if } x \in [i, i+1]$$

Define $Y_i \sim_p F(X_i)$ to be the probability integral transform of X_i . From the definition of Y_i , we have that $\mathbb{P}(x_i \in [i, i+1]) = \mathbb{P}(y_i \in [0, \frac{l_i}{L}])$. It follows that if \mathbf{X} is connected, then \mathbf{Y} lies within \mathcal{C}' . Moreover, \mathbf{Y} is jointly uniform on the half-space

$$(5.9) \quad \mathcal{T}' = \{\mathbf{y} \in \mathbb{R}^n : \mathbf{y} \geq 0 \text{ and } \sum y_i \leq s\}.$$

Thus, $p_{\text{con}} = \text{Vol}(\mathcal{T}' \cup \mathcal{C}') / \text{Vol}(\mathcal{C}')$, from which the solution to VHSP can be computed in P time. \square

5.2.2. Extensions of Theorem 5.3. We may extend Theorem 5.3 to more general parents satisfying the constraints that:

$$(5.10) \quad \int_{(i-1)d}^d {}_p f(x) dx = l_i, \text{ for all } i = 1, \dots, n, \text{ where } l_i \geq 0 \text{ and } \sum l_i = 1.$$

Since Eq.(5.10) provides us with n constraint equations, ${}_p f$ needs to have at least n parameters to fit them, e.g. polynomials of degree n , with arbitrary coefficients. More generally, if f_1, \dots, f_n are arbitrary pdfs with unit supports, each having at least one parameter, their mixture

$$(5.11) \quad {}_p f(x) = f_i(x) \text{ if } x \in [i-1, i)\mathbf{I}_{\mathcal{B}}, \text{ where } \mathcal{B} = [0, n+1]$$

may be fit to obey Eq.(5.10). Consequently, computing PCON for this mixture is #PH.

On the other hand, problems with a constant number of parameters fail to admit such a reduction analogous to Theorem 5.3, fail to be #PH even though they may not exhibit an explicit formula for #PH. For example, we do not know a short formula for computing p_{con} for Renyi parking, as mentioned in Sec.4.1. Nonetheless, the problem lacks sufficiently many free coefficients to admit a reduction, and consequently is not #PH. Likewise is the case of nonuniform parents such as Beta, Triangular, and clipped Gaussian pdfs on \mathcal{B} .

5.2.3. Robots with heterogeneous connectivity thresholds. We now consider a *Heterogeneous* robot team whose Wifi adapters have different transmission power, so that \mathcal{R}_i has connectivity threshold d_i . We call \mathcal{R}_i *weaker* (resp. *stronger*) than \mathcal{R}_j iff $d_i < d_j$ (resp. $d_i > d_j$). If $d_i = d_j$ we say that the two robots have equal power. Then the robot network is represented by the digraph \mathcal{G} , whose directed edges are of the form $i \rightarrow j$ iff \mathcal{R}_i can transmit to \mathcal{R}_j . In general, edges are not bidirectional, since a weaker robot will not sense a stronger

one, even though the converse holds true. Suppose without losing generality that the d_i 's form a strictly positive, non-decreasing sequence.

Consider a configuration in which the robots are arranged from left to right in increasing order of their index. A connected configuration satisfies the $n + 1$ constraints

$$(5.12) \quad s_{2i-1}, s_{2i} \leq d_i \text{ for all } i : 1, \dots, \lfloor n/2 \rfloor, \text{ and in addition} \\ s_{n+1} \leq d_{n/2} \text{ if } n \text{ is even.}$$

In general, a connected configuration will have two distinct slacks s_{i1}, s_{i2} each less than d_i . Further, the connected region is the intersection of \mathcal{S} with the union of hypercuboids, each of which has two dimensions equal to d_i , and an extra dimension equal to $d_{n/2}$ if n is even. Define \mathcal{H} to be:

$$(5.13) \quad \mathcal{H} = \bigcup \prod [0, a_i], \text{ where the } a_i \text{ are a permutation of the } d_i.$$

The number of hypercuboids in \mathcal{H} is at most $n!$, which is the case when all d_i 's are distinct. We will assume that the d_i 's are distinct unless mentioned otherwise. Since \mathcal{H} is nonconvex in general, so is $\mathcal{F} = \mathcal{H} \cap \mathcal{S}$. When s is sufficiently large that all n robots are required to connect it, \mathcal{F} becomes the disjoint union of $n!$ pieces, each of which is the intersection of \mathcal{S} with one of the component hypercuboids of \mathcal{H} . Then we have that $\text{Vol}(\mathcal{F}) = n! \text{Vol}(\mathcal{S} \cap \mathcal{C}')$, where \mathcal{C}' is the hypercube with dimensions $d_1 \times d_1 \dots \times d_n \times d_n$.

THEOREM 5.4. *PCON is #PH for heterogeneous connectivity.*

Proof. Consider the odd-numbered instance of VHSP in dimension $2n + 1$, with hypercuboid dimensions $l_1, l_1, \dots, l_n, l_n, l_{n+1}$ and slack sum b . Assume that the l_i 's are distinct. It is clear that this instance of VHSP is at least as hard as its counterpart in \mathbb{R}^n with dimensions l_1, \dots, l_n , and thus is #PH. The equivalent instance of $s = \frac{b}{L} + 1$ and $(d_i)_{1 \leq i \leq n} = l_i/L$, where $L = \sum l_i$ as before. The solution of VHSP is now $n! p_{\text{con}} \text{Vol}(\mathcal{C}')$, which is computable in P time from the solution to PCON.

□

It is immediately clear that finding $\text{Vol}(\mathcal{F})$ and $\text{Vol}(\mathcal{U})$ is #PH for heterogeneous connectivity. With homogeneous robots, \mathcal{F} was more symmetric compared to its heterogeneous counterpart. Exploiting this symmetry led to relatively short formulae for PCON and the like. On the other hand a heterogeneous swarm is sufficiently diverse that its connected region be an arbitrary half-space. We pay for this expressiveness by making the connectivity problems harder. Computing p_{con} has a Fully Polynomial Randomized Approximation Scheme (FPRAS), which samples a uniform pdf over a subset of \mathcal{F} in P, using the Markov Chain Monte Carlo (MCMC) method [16]. Combining MCMC with Inverse CDF Sampling enables us to sample arbitrary IID pdfs over \mathcal{F} . This approach is sufficiently general that it adapts to arbitrary joint pdfs over \mathcal{F} , in which case it becomes the Metropolis-Hastings sampling [3].

5.3. Inid parents. We will finally relax the iid assumption by assuming that position X_i has the parent pdf p_{fi} , and is chosen independently of others. We denote the vector of parent pdfs and cdfs by ${}_p\mathbf{f}_{1:n}$ and ${}_p\mathbf{F}_{1:n}$ respectively.

THEOREM 5.5. *PCON-inid, the version of PCON generated by the inid parents $\mathbf{X} \sim {}_p\mathbf{f}_{1:n}$, where each parent is supported on \mathcal{B} , is #PH.*

Proof. We reduce a #PH subset of PCON(nU) to PCON-inid. Let the given instance of PCON(nU) have the boundary length $s = n + 1$, and the parent defined by the $n + 1$ pieces $pf_{i:1\dots n} = p_i$ over $[i - 1, i]$, with $\sum p_i = 1$. The equivalent instance of PCON-inid will have $2U$ parents and a boundary length $s' = n + 2$. Define the i -th parent pf'_i of the inid instance to be

$$(5.14) \quad pf'_i = \begin{cases} p_i & \text{for } x \in [i - 1, i] \\ 1 - p_i & \text{for } x \in [n + 1, n + 2] \\ 0 & \text{elsewhere on } [0, n + 2]. \end{cases}$$

It is clear that pf'_i has unit measure on its support, and is thus a pdf. Further, if the original nU instance is connected, then so is the inid instance *on the interval* $[0, n + 1]$, with connectivity on the last segment $[n + 1, n + 2]$ *ignored*. The unrestricted connectivity of the last segment has no effect on the complexity of the problem. \square

REFERENCES

- [1] MARK DE BERG, OTFRIED CHEONG, MARK VAN KREVELD, AND MARK OVERMARS, *Computational Geometry: Algorithms and Applications*, Springer-Verlag TELOS, Santa Clara, CA, USA, 3rd ed., 2008.
- [2] SPRING BERMAN, ÁDÁM HALÁSZ, M. ANI HSIEH, AND VIJAY KUMAR, *Optimized stochastic policies for task allocation in swarms of robots*, IEEE Trans. Robot., 25 (2009), pp. 927–937.
- [3] SIDDHARTHA CHIB AND EDWARD GREENBERG, *Understanding the Metropolis-Hastings algorithm*, The American Statistician, 49 (1995), pp. 327–335.
- [4] NIKOLAUS CORRELL, *Parameter estimation and optimal control of swarm-robotic systems: a case study in distributed task allocation*, in Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, May 19–23, 2008, pp. 3302–3307.
- [5] H.A. DAVID AND H.N. NAGARAJA, *Order Statistics*, John Wiley and Sons, Hoboken, NJ, USA, 2003.
- [6] A. DVORETZKY AND H. ROBBINS, *On the parking problem*, Publications of the Mathematical Institute of the Hungarian Academy of Sciences, (1964), pp. 209 – 224.
- [7] MARTIN DYER AND ALAN FRIEZE, *Computing the volume of convex bodies: a case where randomness provably helps*, Probabilistic combinatorics and its applications, 44 (1991), pp. 123–170.
- [8] M. E. DYER AND A. M. FRIEZE, *On the complexity of computing the volume of a polyhedron*, SIAM J. Comput., 17 (1988), pp. 967–974.
- [9] WILLIAM C. EVANS, GRÉGORIE MERMOUD, AND ALCHERIO MARTINOLI, *Comparing and modeling distributed control strategies for miniature self-assembling robots*, in Proceedings of the 2010 IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, May 3–7, 2010, pp. 1438–1445.
- [10] DEBORAH H. GLUECK, ANIS KARIMPOUR-FARD, JAN MANDEL, LARRY HUNTER, AND KEITH E. MULLER, *Fast computation by block permanents of cumulative distribution functions of order statistics from several populations*, Communications in Statistics-Theory and Methods, 37 (2008), pp. 2815–2824.
- [11] NICK GRAVIN, DMITRII V. PASECHNIK, BORIS SHAPIRO, AND MICHAEL SHAPIRO, *On moments of a polytope*, arXiv preprint arXiv:1210.3193, (2012).
- [12] MARTIN HAENGGI, *Stochastic Geometry for Wireless Networks*, Cambridge University Press, 2012.
- [13] INTERNET ENGINEERING TASK FORCE, *Optimized link state routing: RFC 7181*, <http://tools.ietf.org/html/rfc7181>, 2014.
- [14] PHILIPPE JACQUET, PAUL MÜHLETHALER, THOMAS CLAUSEN, ANIS LAOUITI, AMIR QAYYUM, AND LAURENT VIENNOT, *Optimized link state routing protocol for ad hoc networks*, in Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century. Proceedings. IEEE International, IEEE, 2001, pp. 62–68.
- [15] ABBAS JAMALIPOUR AND YAOZHU MA, *Intermittently Connected Mobile Ad Hoc Networks: from Routing to Content Distribution*, (SpringerBriefs in Computer Science), Springer Science, 2011.
- [16] MARK JERRUM, *Counting, Sampling, and Integrating: Algorithms and Complexity*, Lectures in Mathematics, Birkhauser Verlag, 2003.
- [17] ERIC KLAVINS, SAMUEL BURDEN, AND NILS NAPP, *Optimal rules for programmed stochastic self-assembly*, in Proceedings of Robotics: Science and Systems, Philadelphia, PA, USA, August 2006.
- [18] GANESH P. KUMAR AND SPRING BERMAN, *Analytical results and algorithms for stochastic boundary coverage by robotic swarms*, Submitted to IEEE Transactions on Robotics, (2014).

- [19] GANESH P. KUMAR AND SPRING BERMAN, *Statistical analysis of stochastic multi-robot boundary coverage*, in Int'l. Conf. on Robotics and Automation (ICRA), 2014, pp. 74–81.
- [20] GANESH P. KUMAR, AURÉLIE BUFFIN, THEODORE P. PAVLIC, STEPHEN C. PRATT, AND SPRING M. BERMAN, *A stochastic hybrid system model of collective transport in the desert ant *Aphaenogaster cockerelli**, in Proc. 16th Int'l. Conf. on Hybrid Systems: Computation and Control (HSCC), 2013, pp. 119–124.
- [21] WENGUO LIU AND ALAN F. T. WINFIELD, *Modeling and optimization of adaptive foraging in swarm robotic systems*, Int. J. Robot. Res., 29 (2010), pp. 1743–1760.
- [22] T. WILLIAM MATHER AND M. ANI HSIEH, *Distributed robot ensemble control for deployment to multiple sites*, in Proceedings of Robotics: Science and Systems, Los Angeles, CA, USA, June 2011.
- [23] LOIČ MATTHEY, SPRING BERMAN, AND VIJAY KUMAR, *Stochastic strategies for a swarm robotic assembly system*, in Proceedings of the 2009 IEEE International Conference on Robotics and Automation, Kobe, Japan, May 12–17, 2009, pp. 1953–1958.
- [24] GREG MULLER, *The efficiency of random parking*. <http://tiny.cc/7r8a7x>, 2007.
- [25] NILS NAPP, SAMUEL BURDEN, AND ERIC KLAVINS, *Setpoint regulation for stochastically interacting robots*, in Proceedings of Robotics: Science and Systems, Seattle, WA, USA, June 2009.
- [26] LAEL U. ODHNER AND HARRY ASADA, *Stochastic recruitment control of large ensemble systems with limited feedback*, J. Dyn. Syst., Meas., Control., 132 (2010).
- [27] MATTHEW PENROSE, *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, 2003.
- [28] ———, *Random Geometric Graphs*, Oxford Studies in Probability, Oxford University Press, 2003.
- [29] ALFRÉD RÉNYI, *On a one-dimensional problem concerning random space-filling*, Publ. Math. Inst. Hung. Acad. Sci., 3 (1958), pp. 109–127.
- [30] MATHIEU DUTOUR SIKIRIC AND YOSHIAKI ITOH, *Random Sequential Packing of Cubes*, World Scientific Publishing Company, 2011.
- [31] J TALBOT, G TARJUS, PR VAN TASSEL, AND P VIOT, *From car parking to protein adsorption: an overview of sequential adsorption processes*, Colloids and Surfaces A: Physicochemical and Engineering Aspects, 165 (2000), pp. 287–324.
- [32] CHEN WANG, GUANGMING XIE, AND MING CAO, *Controlling anonymous mobile agents with unidirectional locomotion to form formations on a circle*, Automatica, 50 (2014), pp. 1100 – 1108.